IN THE SPECIFICATION:

Please amend the specification as follows:

Please replace the following paragraphs on pages 1 through 4 beginning at page 1, line 6, with the following new paragraphs:

Provisional U.S. Patent Application No. 60/076,048, entitled "Distributed Computing System" and filed on February 26, 1998.

U.S. Patent Application No. 09/044,923, entitled "Method and System for Leasing Storage," filed on March 20, 1998, now U.S. Patent No. 6,263,350.

U.S. Patent Application No. 09/044,838, entitled "Method, Apparatus, and Product for Leasing of Delegation Certificates in a Distributed System," filed on June 26, 2000, and now U.S. Patent No. 6,247,026.

U.S. Patent Application No. 09/044,834, entitled "Method, Apparatus and Product for Leasing of Group Membership in a Distributed System," filed on March 20, 1998, and now 6,421,704.

U.S. Patent Application No. 09/044,916, entitled "Leasing for Failure Detection," filed on March 20, 1998, and now U.S. Patent No. 6,016,500.

U.S. Patent Application No. 09/044,933, entitled "Method for Transporting Behavior in Event Based System," filed on March 20, 1998, and now U.S. Patent No. 6,463,446.

U.S. Patent Application No. 09/044,919, entitled "Deferred Reconstruction of Objects and Remote Loading for Event Notification in a Distributed System," filed on March 20, 1998, and now U.S. Patent No. 6,272,559.

U.S. Patent Application No. 09/044,938, entitled "Methods and Apparatus for Remote Method Invocation," filed on March 20, 1998, and now U.S. Patent No. 6,487,607.

U.S. Patent Application No. 09/045,652, entitled "Method and System for Deterministic Hashes to Identify Remote Methods," filed on March 20, 1998, and now U.S. Patent No. 6,134,603.

U.S. Patent Application No. 09/044,790, entitled "Method and Apparatus for Determining Status of Remote Objects in a Distributed System," filed on March 20, 1998, and now U.S. Patent No. 6,598,094.

U.S. Patent Application No. 09/044,930, entitled "Downloadable Smart Proxies for Performing Processing Associated with a Remote Procedure Call in a Distributed System," filed on March 20, 1998, and now U.S. Patent No. 6,393,497.

U.S. Patent Application No. 09/044,917, entitled "Suspension and Continuation of Remote Methods," filed on March 20, 1998, and now U.S. Patent No. 6,237,024.

U.S. Patent Application No. 09/044,835, entitled "Method and System for Multi-entry and Multi-Template Matching in a Database," filed on March 20, 1998, and now U.S. Patent No. 6,182,083.

U.S. Patent Application No. 09/044,839, entitled "Method and System for In-Place Modifications in a Database," filed on March 20, 1998, and now abandoned.

U.S. Patent Application No. 09/044,945, entitled "Method and System for Typesafe Attribute Matching in a Database," filed on March 20, 1998, and now U.S. Patent No. 6,578,044.

U.S. Patent Application No. 09/044,931, entitled "Dynamic Lookup Service in a Distributed System," filed on March 20, 1998, and now U.S. Patent No. 6,185,611.

U.S. Patent Application No. 09/044,939, entitled "Apparatus and Method for Providing Downloadable Code for Use in Communicating with a Device in a Distributed System," filed on March 20, 1998, and now U.S. Patent No. 6,560,656.

U.S. Patent Application No. 09/044,932, entitled "Apparatus and Method for Dynamically Verifying Information in a Distributed System," filed on March 23, 2001, and now U.S. Patent No. 6,466,947.

U.S. Patent Application No. 09/030,840, entitled "Method and Apparatus for Dynamic Distributed Computing Over a Network," filed on February 26, 1998, and now U.S. Patent No. 6,446,070.

U.S. Patent Application No. 09/044,936, entitled "An Interactive Design Tool for Persistent Shared Memory Spaces," filed on June 20, 2000, and now abandoned.

U.S. Patent Application No. 09/044,934, entitled "Polymorphic Token-Based Control, " filed on April 27, 2001, and now U.S. Patent No. 6,438,614.

U.S. Patent Application No. 09/044,915, entitled "Stack-Based Access Control," filed on March 20, 1998, and now U.S. Patent No. 6,138,238.

U.S. Patent Application No. 09/044,944, entitled "Stack-Based Security Requirements," filed on March 20, 1998, and now U.S. Patent No. 6,226,746.

U.S. Patent Application No. 09/044,837, entitled "Pre-Method Designation of Security Requirements," filed on June 16, 2000, and now U.S. Patent No. 6,282,652.

On page 26, amend the paragraph starting on line 7 with the following new paragraph:

The lookup service may contain a subset of all services available in the network, referred to as a "Djinn" as described in copending U.S. Patent Application Serial No. 09/044,931, entitled "Dynamical Lookup Service in a Distributed System," assigned to a common assignee, filed on March 20, 1998 even date herewith, now U.S. Patent No. 6,185,611, which has been previously incorporated by reference. A "Djinn" refers to a logical grouping of one or more of the services or resources that are provided by a network. Devices connected to the network may either dynamically add themselves to the Djinn or dynamically remove themselves from the Djinn. When added, a device provides zero or more of its services to Djinn and may utilize all of the services currently provided by the Djinn. The services provided by the Djinn are defined by the lookup service, which provides a common way to both find and utilize the services for the Djinn.

On page 27, please amend the paragraph starting on line 12 with the following new paragraph:

When a new service is created (e.g., when a new device is added to the Djinn), the service registers itself with the lookup service 400, providing a stub 404 to be used by a client to access the service and an initial collection of attributes 406 associated with the service. For example, a printer might include attributes indicating speed (in pages per minute), resolution (in dots per inch), color, and whether duplex printing is supported. The lookup service administrator (not shown) might also add new attributes, such as the physical location of the service and common names for it. Additionally, if a

service encounters some problem that needs administrative attention, such as a printer running out of toner, the service can add an attribute that indicates what the problem is. In one implementation consistent with the present invention, attributes are stored as multi-entries, and the addition, modification and deletion of attributes can be made using multi-templates and the techniques explained in co-pending U.S. Patent Application No. 09/044,839, entitled "Method and System for In-Place Modifications in a Database" filed March 20, 1998, now abandoned, and previously incorporated herein.

On page 29, amend the paragraph starting on line 17 with the following new paragraph:

Programs (including other services) that need a particular type of service can use the lookup service 400 to find a stub that can be used to access the service. A match can be made based on the type of service as well as the specific attributes attached to the service. For example, a client could search for a printer by requesting a stub type corresponding to the service desired or by requesting certain attributes such as a specific location or printing speed. In one implementation consistent with the present invention, attributes are stored as multi-entries, and a match on attributes can be made using multi-templates, as explained in co-pending U.S. Patent Application No. 09/044,835, entitled "Method and System for Multi-Entry and Multi-Template Matching In A Database," filed March 20, 1998, now U.S. Patent No. 6,182,083, and previously incorporated herein.

On page 30, please amend the paragraph starting on line 2 with following paragraph:

Referring back to Fig. 4, the stub 404 corresponding to a service is registered in the lookup service 400 and is used by the client computer 11(n) to access the service methods remotely. This stub 404 may be a "smart proxy." A smart proxy, code within which a stub is embedded, helps the client more efficiently implement the stub and the method to be remotely invoked. A smart proxy often performs some local computation for efficiency before and after it actually calls the stub. For example, a smart proxy may contain code to cache information, so if a client requested it again, instead o going back to the server to get the information, it may have cached the answer and be able to return it quickly. If the situation called for it, a smart proxy might also transform the parameters received from the client into other types and then send the transformed types. The smart proxy concept is further explained in co-pending U.S. Patent Application No. 09/044,930, entitled "Downloadable Smart Proxies for Performing Processing Associated with a Remote Procedure Call in a Distributed system, "assigned to a common assignee, filed on March 20, 1998, now U.S. Patent No. 6,393,497 an even date herewith, and which is hereby incorporated by reference.